REMARKS

This communication is responsive to the Final Office Action dated April 28, 2005. Applicants have not amended any of the claims. Claims 1-28 remain pending.

Claim Rejection Under 35 U.S.C. § 102

Rejection of caims 1-7 and 22 in view of Devine

In the Final Office Action, the Examiner rejected claims 1-7 and 22 under 35 U.S.C. 102(e) as being anticipated by Devine et al. (USPN 6,598,167 B2). Applicants respectfully traverses the rejection. Devine et al. (Devine) fails to disclose each and every feature of the claimed invention, as required by 35 U.S.C. 102(c), and provides no teaching that would have suggested the desirability of modification to include such features.

As a preliminary comment, in rejecting Applicants' claims, the Examiner again cites functions provided by a wide number of separate and distinct devices located throughout the Devine system, including web servers 24, application servers 40, dispatcher server 26 and the "HydraWeb Load Balancer 45." As discussed at length in Applicants' previous response, Devine fails to teach or suggest a load balancing acceleration device that comprises both an encryption and decryption engine and a load balancing engine, as required by claim 1. In particular, Devine fails to teach or suggest a device that comprises both (i) an encryption and decryption engine instructing a processor to decrypt data received via the secure communication session and direct the decrypted data to one of a plurality of server devices via a second communication session, and (ii) a load balancing engine associating each of said client devices with a respective one of said server devices based on calculated processing loads of each said server devices. Devine provides no teaching or suggestion of a single device that provides such features.

In contrast, Devine describes a "double firewalled" system having a first firewall, a plurality of web servers, a second firewall and a plurality application servers. None of the devices described by Devine operate as an acceleration device that includes both an encryption and decryption engine and a load balancing engine that associates each of the client devices with a respective server based on calculated processing loads of each server. Rather, the Devine system includes describe a plurality of web servers 24 that essentially act as secure relay devices. More specifically, web servers 24 located within a Demilitarized Zone (DMZ) communicate with client devices via a first set of secure communications sessions (e.g., HTTPS), and relay requests to appropriate application servers 40 located within the enterprise via a dispatcher server 26.

In rejecting claim 1, the Examiner specifically cited column 8, lines 22-65 of Devine. In the cited portions, however, Devine refers to Figure 4 and describes web servers 24 as utilizing SSL and HTTPS to relay communications between client devices and application servers 40 via secure TCP messaging sessions. For example, the cited portion of Devine specifically states that web servers 24 receive messages from client devices S-HTTP or HTTPS, and then forwards the requests to dispatcher server 26 located inside the enterprise Intranet. According to Devine, dispatcher 26 re-encrypts the messages and forwards the messages to the "appropriate" application server based on the service specifically requested by the client. In other words, dispatcher 26 forwards the message to the application server 40 that provides the requested service, e.g., electronic mail, broadband, service inquiries and other services.

Thus, it is clear that neither web servers 24 nor dispatcher 26 of the Devine system performs any form of load balancing. Devine makes clear that the messages are sent to a server that can service the requests, not as function of processing loads. Thus neither web servers 24 nor dispatcher 26 include a load balancing engine that associates each of said client devices with a respective one of said servers based on calculated processing loads of each said server, as required by Applicants' claim 1. Consequently, none of the devices of the Devine system comprises both an encryption and decryption engine for providing secure communications and a load balancing engine, as recited by Applicants' claim 1. In fact, it appears the only device within the Devine system that performs any form of load balancing is the "HydraWeb Load Balancer 45" depicted in Figure 4 and further described in column 23. However, as clearly shown in Figure 4, this device is a dedicated switching unit that does not comprise both an encryption and decryption engine and a load balancing engine, as recited by Applicants' claim 1.

With respect to claim 2, Devine fails to teach or suggest a load balancing acceleration device that comprises both a TCP communications manager and a secure communications manager. In rejecting claim 2, the Examiner refers to col. 23, ln. 17 to col. 24, ln. 14. However, this section of Devine merely describes the conventional HydraWeb Load Balancer 45, which does not perform any form of encryption of decryption. Thus, in rejecting Applicants' claim to an acceleration device, the Examiner appears to erroneously attribute load-balancing functions from the HydraWeb Load Balancer 45 to other devices of the Devine system and vice versa. Neither web servers 24, dispatcher 26 nor the HydraWeb Load Balancer 45 of the Devine system teach or suggest an acceleration device that includes both an encryption and decryption engine for providing secure communications and a load balancing engine, as required by claim 2.

Devine fails to teach or suggest a load balancing acceleration device that comprises a secure communications manager that negotiates a secure communication session with each of said plurality of client devices over an open network, and a TCP communications manager that negotiates a separate, open communications session with one of the plurality of server devices associated with the enterprise for each secure communication session negotiated with the client devices based on the associations of said client devices to said server devices by said load balancing engines, as required by Applicants' claims 3 and 4.

In contrast to the Examiner's assertions, the Devine system fails to describe a device that negotiates secure communication session with client devices and respective open (non-secure) communications sessions to servers. Rather, Devine makes clear that web servers 24 essentially act as secure relay devices. More specifically, web servers 24 located within a Demilitarized Zone (DMZ) communicate with client devices via a first set of secure communications sessions (e.g., HTTPS), and relay communications to appropriate application servers 40 located within the enterprise via a dispatcher server 26 and a second set of secure communications.

Thus, the Examiner's analysis in rejecting in claims 3 and 4 is flawed for at least two reasons. First, in the Devine system, communications are relayed using two sets of secure communication sessions. Thus, no device in the Devine system is actually operating as an acceleration device on behalf of another device and, therefore, no device negotiates secure communication sessions with the client devices and negotiates open communications session with server devices, as required by Applicants' claims. Second, there is no indication of a one-to-one mapping in the Devine system between secure communication sessions from the client devices to the acceleration device and the open (non-secure) communication sessions from the acceleration device to the server device as required by Applicants' claims 4 and 9.

Applicants' claim 5 requires that the claimed acceleration device performs encryption and decryption on the packet level by decrypting packet data received via the secure communication session to extract a secure record, decrypting application data from the secure record in the packet data, and outputting the decrypted application data from the secure record to the one of said server devices via the second communication session without processing the application data with an application layer of a TCP/IP stack.

For purposes of clarity, Applicants refer the Examiner to Figure 2B and pages 5 and 6 that describe conventional SSL acceleration devices and, in particular, how in prior art systems HTTP packets conventionally traverse the entire networking protocol stack including the IP layer, the SSL session layer and the application layer multiple times.

In contrast, embodiments of the present invention include an acceleration device that operates at the packet level. Applicants refer the Examiner to page 10, ll. 3-13 of the present application that states:

> *Figure 3 shows how the system of the present invention differs in general from that of the prior art, and illustrates the manner in which the SSL encryption and decryption proxy is implemented. Typically, when a Web client wishes to send data via a secure protocol to an SSL enabled Web server, it will do so by communicating via a secure port 443. As shown in Figure 3, in accordance with the present invention, the SSL accelerator will intercept data destined for port 443 of the web server and, rather than the transmitting packets up and down the TCP/IP stack as shown in Figure 2B, will perform the SSL encryption and decryption at the packet level before forwarding the packet on to its destination. The accelerator will thus decode the packet data and forward a clear text (HTTP) packet the HTTP port 80 of the Web server 300.*

Further, on page 16, ll. 17-26, the present application states that:

> *As shown at reference number 265, client 100 will now begin sending encrypted application data to the SSL accelerator device 250. ... The accelerator device will process the data at step 270 on the packet level and forward it to the server as clear text.*

Devine fails to teach or suggest an acceleration device having an encryption and decryption engine that decrypts the data on a packet level by decrypting packet data received via the secure communication session to extract an SSL record, decrypting application data from the SSL record in the packet data, and outputting the decrypted application data from the SSL record to the one of said server devices via the second communication session without processing the application data with an application layer of a TCP/IP stack, as required by claim 5.

In contrast, the Devine system describes web servers 24 as implementing HTTPS, which operates at the application layer (HTTP) and the session layer (SSL). Devine specifically describes decrypting and re-encrypting messages and forwarding the messages via HTTP. Thus, Devine does not describe an SSL acceleration device that operations on the packet-level as described and claimed by the Applicants.

With respect to claim 22, Devine fails to teach or suggest a network router that comprises both an encryption and decryption engine and a load balancing engine. In rejecting claim 22, the Examiner refers to col. 22, lines 47-65. However, in the cited passage Devine refers to routers 29(a), (b) of FIGS. 1, 4 and 5. Thus, the Examiner refers to yet another device distinct from the other devices referred to above. In summary, in rejecting Applicants' claim 22, the Examiner has incorrectly cited functions provided by many separate devices in the Devine system.

Devine et al. fails to disclose each and every limitation set forth in claims 1-7 and 22. For at least these reasons, the Examiner has failed to establish a prima facie case for anticipation of Applicants' claims 1-7, 9 and 10 under 35 U.S.C. 102(e). Withdrawal of this rejection is requested.

Rejection of claims 12-15 and 17-24 in view of Huppenthal

In the Final Office Action, the Examiner also rejected claims 12-15 and 17-24 under 35 U.S.C. 102(e) as being anticipated by Huppenthal et al. (USPN 6,434,687 B1). Applicants respectfully traverses the rejection. Huppenthal et al. (Huppenthal) fails to disclose each and every feature of the claimed invention, as required by 35 U.S.C. 102(e), and provides no teaching that would have suggested the desirability of modification to include such features.

As a preliminary comment, Huppenthal describes a multiprocessor computing architecture that utilizes reconfigurable microprocessors. That is, Huppenthal describes a multi-processor architecture that may be used within a single server, such as a web server. FIG. 1 of Huppenthal shows a multi-processor architecture utilizing N processors connected to M memory banks via a memory interconnect fabric. Huppenthal is completely and entirely unrelated to Applicants' claimed invention directed to an intermediate acceleration device that that interfaces between client devices and server devices. The Examiner appears to be confusing the internal processors of the Huppenthal architecture with different devices within a network.

With respect to claim 12, Huppenthal fails to teach or suggest nearly every single element of Applicants' claim. For example, Huppenthal fails to teach or suggest an intermediate acceleration device that receives communications from customer devices, decrypts the communications and forwards the communications to a server. The portions of the Huppenthal cited by the Examiner refer to "decoding" instructions for execution by the processors, which has nothing to do with "decrypting" secure communications from a client device. Huppenthal makes passing statements that the multiprocessor architecture may be used to improve processing of secure sockets, but this merely refers to an improved processing speed of the server. Thus, the Examiner appears to be construing the Huppenthal architecture as an intermediate acceleration device solely because the architecture may increase the speed or processing secure data. This logic is flawed in that Huppenthal is not describing an intermediate acceleration device that receives communications from customer devices, decrypts the communications and forwards the communications to a server, as specifically recited by claim 12.

Moreover, Huppenthal fails to describe any load-balancing functions across servers as recited by claim 12. For example, Huppenthal fails to teach or suggest selecting with the acceleration device at least one of the plurality of servers in the enterprise based on a load calculation including processing sessions of other servers in the enterprise and associating the selected server with a communications session from the one of the clients, as claimed. In reference to these functions, the Examiner cites col. 7, ll. 1-25 of Huppenthal that describes how executable instructions are processed by the multiprocessor architecture and that programs may be transferred to a different one of the N microprocessors. This is completely unrelated to load balancing functions recited by Applicants' claim 12. First, Huppenthal does not describe selecting one of a plurality of servers. Presumably, the Examiner has mistaken processors for servers within the Huppenthal architecture. Moreover, Huppenthal fails to teach or suggest selecting servers based on a load calculation including the processing sessions of the servers.

In addition, Huppenthal does not describe forwarding the decrypted packet data from the acceleration device to the selected server of the enterprise. With respect to these elements, the Examiner again cites col. 7, ll. 1-25 of Huppenthal that describes how executable instructions are processed by the multiprocessor architecture and that programs may be transferred to a different one of the N microprocessors. This is entirely unrelated to forwarding packets at all, let alone forwarding packets from an intermediate acceleration device to a server.

With respect to claim 13, it is apparent that Huppenthal fails to teach or suggest encrypting the application data received from the selected server, and forwarding the encrypted application data to a customer device. With respect to claim 13, the Examiner again cites col. 7, ll. 1-25 of Huppenthal that describes how executable instructions are processed by the multiprocessor architecture and that software programs may be transferred to a different one of the N microprocessors. This is completely unrelated forwarding application data to a customer device. Apparently now the Examiner has misunderstood the microprocessors of Huppenthal to be customer devices.

With respect to claim 14, Huppenthal fails to teach or suggest receiving with the device communications having a destination IP address of the enterprise. In the portion cited by the Examiner, Huppenthal describes an input buffer and an output FIFO for interfacing with a memory. Applicants are left to wonder what relevance interfacing with memory this has at all to the claimed step of receiving application data from a server by receiving a communication having a destination IP address of the enterprise. The cited portion of Huppenthal does not describe receiving a network communication or network addresses at all, let alone a communication having a destination address of an enterprise.

With respect to claim 15, Huppenthal fails to teach or suggest negotiating the secure protocol session with the customer device by responding as the enterprise to the customer devices. In rejecting claim 15, the Examiner again cites col. 7, ll. 1-25 of Huppenthal that describes how executable instructions are processed by the multiprocessor architecture and that programs may be transferred to a different one of the N microprocessors. This is entirely unrelated to negotiating the secure protocol session with the customer device, let alone responding as the enterprise to the customer devices. Applicants can find no relevance of Huppenthal.

Similarly, with respect to claim 17, Huppenthal fails to teach or suggest establishing an open communication session from the acceleration device to the selected server, and mapping the decrypted packet data to the open communication session established with the selected server. In rejecting claim 17, the Examiner again cites col. 7, ll. 1-25 of Huppenthal that describes how executable instructions are processed by the multiprocessor architecture and that programs may be transferred to a different one of the N microprocessors. This is entirely unrelated to establishing open communication session. Further, processing instructions with a multi-processor architecture is unrelated to mapping decrypted packet data to the open communication session established with the selected server. Again, Applicants can find no relevance of the cited portions of Huppenthal.

Similarly, with respect to claim 18, Huppenthal fails to teach or suggest that the open communication session is established via a secure network. In rejecting claim 17, the Examiner again cites col. 7, ll. 1-25 of Huppenthal that describes how executable instructions are processed by the multiprocessor architecture and that programs may be transferred to a different one of the N microprocessors. This teaching is unrelated to establishing open communication session within a secure network.

With respect to claim 19, Huppenthal fails to teach or suggest receiving encrypted data having a length greater than a TCP segment carrying said data, and wherein said step of decrypting comprises: buffering the encrypted data in a memory buffer in the accelerator device, the buffer having a length equivalent to the block cipher size necessary to perform the cipher; and decrypting the buffered segment of the received encrypted data to provide decrypted application data. Contrary to the Examiner's assertion, Huppenthal makes no mention of these novel features. In fact, the first cited portion of Huppenthal (col. 8, ll. 47-67) describe an address generator for reading data from a buffer within the Huppenthal multi-processor architecture. The second cited portion of Huppenthal (col. 7, ll. 1-25) describe how executable instructions are processed by the multiprocessor architecture and that programs may be transferred to a different one of the N microprocessors. These passages are completely unrelated to Applicants' claim 19 which require specific techniques for decrypting segments of encrypted data. Huppenthal does not describe a block cipher size nor the relation of a buffer size with respect to the block cipher size.

Applicants' claim 20 requires that the data is authenticated on a packet level on receipt of a final TCP segment without processing the application data with an application layer of a TCP/IP stack. It appears the Examiner has overlooked or misunderstood these requirements.

The portion of Huppenthal cited by the Examiner (col. 9, ll. 43-67) describes storage of output from the Huppenthal multi-processor architecture. This is entirely unrelated to authenticating of a packet, let alone authenticating without processing application data at an application layer. Huppenthal does not describe any of these elements.

With respect to claim 24, Huppenthal fails to describe decrypting the data packets to extract a secure record. Huppenthal fails to describe decrypting application data from the secure record. And, Huppenthal fails to describe authenticating the application data without processing the application data with an application layer of a TCP/IP stack. With respect to all of these elements, the Examiner yet again cites col. 7, ll. 1-25 of Huppenthal that describes how executable instructions are processed by the multiprocessor architecture and that programs may be transferred to a different one of the N microprocessors. This is unrelated to functions recited by Applicants' claim 24.

In order to support an anticipation rejection under 35 U.S.C. 102(e), it is well established that a prior art reference must disclose each and every element of a claim. This well known rule of law is commonly referred to as the "all-elements rule."[1] If a prior art reference fails to disclose any element of a claim, then rejection under 35 U.S.C. 102(e) is improper. [2] Huppenthal fail to disclose each and every limitation set forth in claims 12-15 and 17-24. For at least these reasons, the Examiner has failed to establish a prima facie case for anticipation of Applicants' claims 12-15 and 17-24 under 35 U.S.C. 102(e). Withdrawal of this rejection is requested.

Rejection of claims 25-28 in view of Baskey

In the Final Office Action, the Examiner rejected claims 25-28 under 35 U.S.C. 102(e) as being anticipated by Baskey et al. (USPN 6,732,269 B1). Applicants respectfully traverse the rejection. Baskey et al. (Baskey) fails to disclose each and every feature of the claimed invention, as required by 35 U.S.C. 102(e), and provides no teaching that would have suggested the desirability of modification to include such features.

[1] *See Hybritech Inc. v. Monoclonal Antibodies, Inc.*, 802 F.2d 1367, 231 USPQ 81 (CAFC 1986) ("it is axiomatic that for prior art to anticipate under 102 it has to meet every element of the claimed invention").

[2] *Id. See also Lewmar Marine, Inc. v. Barient, Inc.* 827 F.2d 744, 3 USPQ2d 1766 (CAFC 1987); *In re Bond*, 910 F.2d 831, 15 USPQ2d 1566 (CAFC 1990); *C.R. Bard, Inc. v. MP Systems, Inc.*, 157 F.3d 1340, 48 USPQ2d 1225 (CAFC 1998); *Oney v. Ratliff*, 182 F.3d 893, 51 USPQ2d 1697 (CAFC 1999); *Apple Computer, Inc. v. Articulate Systems, Inc.*, 234 F.3d 14, 57 USPQ2d 1057 (CAFC 2000).

With respect to claim 25, Baskey fails to teach or suggest several elements. In general, Baskey recited by the Examiner describe an SSL proxy server 40 that acts as a proxy server for a transaction server 50. In col. 6, ll. 17-35, Baskey indeed makes mention that the SSL proxy server 40 may serve multiple transaction servers 50. However, Baskey clearly states that in that case, the routing function would determine the destination of a communication from a client route the communication to the corresponding persistent secure connection (col. 6, ll. 32-36). Thus, Baskey makes clear that the destination is selected based on the communication from the client. For example, a destination address may be used, as is typically used with routing functions.

This point servers to illustrate a deficiency in the Examiner's analysis. That is, Baskey fails to teach or suggest that the intermediate device selects one of the server devices based on resource loading experienced by the server devices. Baskey does not describe an intermediate device that selects one of the server devices based on resource loading experienced by the server devices. Baskey provides no teaching or suggestion of consideration for the resource loading experienced by different server devices when selecting one of the servers for which to establish a communication session based on an intercepted request from a client.

In reference to these elements of claim 25, the Examiner cited Baskey at col. 5, ll. 58 – col. 6, ll. 16. However, in these passages, Baskey describes that an SSL proxy server may be capable of processing multiple SSL connections from different clients using a single SSL connection to transaction server 50. Baskey describes the connection to server 50 as a persistent multiplexed SSL connection 40, which is entirely different from load balancing across different servers based on processing loads at the servers. The Applicants refer the Examiner to FIG. 4 of Baskey which is described by the passage cited by the Examiner. FIG. 4 of Baskey clearly illustrates the multiple client connections 36 and 36' that are multiplexed through a single SSL connection 44 to transaction server 50. This is entirely different from an intermediate device that selects one of a plurality of server devices based on resource loading experienced by each of the server devices, and establishing an open communication sessions with the selected server device, as claimed by the Applicants.

With respect to claim 26, Baskey fails to teach an intermediate device that decrypts data from a secure connection and forwards the decrypted data to the selected server device via the non-secure communication session. The portion of Baskey cited by the Examiner (col. 8, ln. 51 – col. 9, ln. 19) makes clear that "[i]n any event, the information connect of the message received over the client specifc SSL connection from the client may then be forwarded over the pseristent SSL connection 44." Thus, Baskey does not describe forwarding decrypted data over a non-secure communication session. It appears that the Examiner may have been referring to the second connection 60 descirbed by Baskey. However, that connections is used to send a "message identifier," not the original content received from the client device.

Similarly, with respect to claim 27, the portion of Baskey cited by the Examiner fails to teach or suggest an intermediate device that receives unencrypted data from the selected server device via the non-secure communication session, encrypts the data and forwards the encrypted data to the client device via the secure communication session.

Baskey fails to disclose each and every limitation set forth in claims 25-28. For at least these reasons, the Examiner has failed to establish a prima facie case for anticipation of Applicants' claims 25-28 under 35 U.S.C. 102(e). Withdrawal of this rejection is requested.

Claim Rejection Under 35 U.S.C. § 103

In the Office Action, the Examiner rejected claims 8 and 11 under 35 U.S.C. 103(a) as being unpatentable over Devine in view of Gelman et al. (US 6,415,329 B1), and rejected claim 16 under 35 U.S.C. 103(a) as being unpatentable over Huppenthal in view Gelman. Applicants respectfully traverse the rejection. The cited references, either singularly or in combination, fail to disclose or suggest the inventions defined by Applicants' claims, and provide no teaching that would have suggested the desirability of modification to arrive at the claimed invention.

In general, Gelman describes a method of communicating over a satellite or other high delay-bandwidth link. Gelman describes modifying destination address in a first protocol, forwarding the packets in a second protocol and then restoring the destination address (Abstract).

With respect to claims 8, 11 and 16, the Examiner states that it would have been obvious to a person having ordinary skill in the art to modify the proxy server of Devine to change the IP addresses. This analysis is flawed in that proxy servers do not modify communication streams. Instead, proxy servers instantiate a communication session with a client and a separate communication session with a second device, such as a server. The communications between the client and the proxy are separate from the communications between the proxy and the server. Typically, different session data is maintained. Thus, the Examiner assertion that a proxy server of Devine could somehow be modified to modify IP addresses of communications flowing through the device to addresses of destination servers. For further clarification, Applicants refer the Examiner to the present application at pg. 12 that describes a "direct, cut through processing method" in which packets from client to server are addressed from the client to the server and from server to client, with the intermediary, SSL device being transparent to both. In this mode, the intermediate device does not act as a proxy device. In contrast, FIG. 7 illustrates the novel acceleration device acting as a complete proxy, in which the device substitutes itself for the server and both the TCP/IP handshaking.

The Examiner's suggestion that it would have been obvious to a person having ordinary skill in the art to modify the proxy server of Devine to change destination IP addresses for the packets to IP addresses for the server devices is incorrect in view of the dual session approach applied by conventional proxy servers, as described by Devine.

For at least these reasons, the Examiner has failed to establish a prima facie case for non-patentability of Applicants' claims 8, 11 and 16 under 35 U.S.C. 103(a). Withdrawal of this rejection is requested.

CONCLUSION

All claims in this application are in condition for allowance. Applicants respectfully request rcconsideration and prompt allowance of all pending claims. Please charge any additional fees or credit any overpayment to deposit account number 50-1778. The Examincr is invited to telephone the below-signed attorney to discuss this application.

Date:

July 26, 2005

SHUMAKER & SIEFFERT, P.A.
8425 Seasons Parkway, Suite 105
St. Paul, Minnesota 55125
Telephone: 651.735.1100
Facsimile: 651.735.1102

By:

Name: Kent J. Sieffert
Reg. No.: 41,312